Steganography and Broadcasting

Fabrice P. Tachago, Stephane G. R. Ekodeck, René Ndoundam

University of Yaounde I, LIRIMA, Team GRIMCAPE, P.o.Box 812 Yaounde, Cameroon

IRD, UMI 209, UMMISCO, IRD France Nord, F-93143, Bondy, France;

Sorbonne Unversités, Univ. Paris 06, UMI 209, UMMISCO, F-75005, Paris, France

E.mail : tachagofabrice@gmail.com, ekodeckstephane@gmail.com, ndoundam@yahoo.com

Abstract

Informally, steganography is the process of exchanging a secret message between two communicating entities so that an eavesdropper may not know that a message has been sent. After a review of some steganographic systems, we found that these systems have some defects. First, there are situations in which some concealment algorithms do not properly hide a secret message. Second, to conceal one bit of a secret message, some ask at least five documents and make at least two sampling operations, thus increasing their run-times. Considering the different ways to communicate with the receiver, we propose two steganographic systems adapted to the email communication whose algorithms are deterministic. To hide one bit of a secret message, our steganographic systems need only one document and performs one sampling operation and therefore significantly reduces the run-time.

Keywords: steganography, run-time, email

1 Introduction

Steganography is the art of concealing secret messages in seemingly innocent media. The use of steganography is not new, as it dates back to antiquity [8, 9, 10]. But scientific study began when G. J. Simmons [1] formulated the problem of communication between two prisoners, which can be stated as follows: Alice and Bob, both prisoners, held in separated cells and remote from one another wish to establish an escape plan without Eve the guardian knowing about it. With Eve allowing them to communicate on condition that the exchanged messages are clear and understandable.

State of art: Two formalisms have been proposed to solve this problem: Theory of Information and Theory of Complexity. In the theory of information, C. Cachin [2] provided a steganographic protocol based on a probabilistic model. It defines the security of a steganographic protocol by a relative entropy between the distribution of covertexts and stegotexts. Here, a protocol is said to be perfectly safe if the relative entropy is null. In the theory of complexity approach, N. Hopper, J. Langford and L. Von Ahn [3], offered a rigorous formalization of a protocol whose security is based on the difficulty for an adversary to distinguish the distribution of covertexts and stegotexts in polynomial time.

N. Hopper and al. [3] proposed a protocol based on a function called rejection sampling. The aim of function is to find in a distribution of covertexts, a covertext which its image by a given pseudo-random function is equal to the secret message's bit that we want to hide. We present now the algorithms proposed in [3].

Algorithm 1 Rejection sampling[3]

Procedure RS

1: **Input**: *Key $K1$, Target x, iteration count, history h*
2: $i = 0$
3: **repeat**
4:    $c \leftarrow \mathbf{S}(h)$
5:    **Increment** i
6: **until** $i = count$ or $F_{K1}(c) = x$
7: **Output:** c

Algorithm 2 Embedding procedure [3]

Procedure $\mathcal{S}1.Embed$

1: **Input**: *Key K, hiddentext $m \in \{0,1\}^*$, history h*
2: **Parse m as** $m_1^1||m_2^1||\cdots||m_n^1$
3: **for** $i = 1\cdots n$ **do**
4:     $c_i = RS(K, m_i, |K|, h)$
5:     $h = h||c_i$
6: **end for**
7: **Output:** $c_1 c_2 \cdots c_n$

Algorithm 3 Extracting procedure [3]

Procedure $\mathcal{S}1.Extract$

1: **Input:** *Key K, stegotexts $c_1 c_2 \cdots c_n$*
2: **for** $i = 1\cdots n$ **do**
3:     $m_i = F_K(c_i)$
4: **end for**
5: $m = m_1||m_2||\cdots||m_n$
6: **Output:** m

The concealment algorithm applies the RS procedure for each bit. The extracting algorithm simply applies pseudo-random function to each document received.

This system possesses two defects: firstly, it needs more sampling operations (expensive operation [7]) which increases its run-time. The algorithm runs in $O(|K|n)$. Secondly this system is not safe for a covertexts distribution which owns a small min-entropy [6].

To correct these flaws, Hopper and al. [4] have proposed another protocol. That protocol is still based on a rejection sampling function. But this time, it makes at most two sampling operations and uses a error correcting code to increase the reliability of the steganographic system.

Algorithm 4 rejection sampling [4]

Procedure RS

1: **Input**: *Key $K1$, Synchronized $N1$, Target x, iteration count, history h*
2: $i = 0$
3: **repeat**
4:     $c \leftarrow \mathbf{S}(h)$
5:     **Increment** i
6: **until** $i = count$ or $F_{K1}(N1, c) = x$
7: **Output**: c

Algorithm 5 Embedding procedure [4]

Procedure $\mathcal{S}2.Embed$

1: **Input**: *Key K, Synchronized N, hiddentext $m' \in \{0,1\}^*$, history h*
2: **Let** $m = Enc(m')$
3: **Parse m as** $m_1^1||m_2^1||\cdots||m_n^1$
4: **for** $i = 1\cdots n$ **do**
5:     $c_i = RS(K, N, m_i, 2, h)$
6:     $h = h||c_i$
7:     **Increment** N
8: **end for**
9: **Output:** $c_1 c_2 \cdots c_n$

Algorithm 6 Extracting procedure [4]

Procedure $S2.Extract$

1: **Input**: *Key K, Synchronized N, stegotexts $c_1 c_2 \cdots c_n$*
2: **for** $i = 1 \cdots n$ **do**
3:     $\tilde{m}_i = F_K(N, c_i)$
4:     **Increment** N
5: **end for**
6: $\tilde{m} = \tilde{m}_1 || \tilde{m}_2 || \cdots || \tilde{m}_n$
7: **Output:** $Dec(\tilde{m})$

The use of a error correcting code is not only time expensive but reduces the transmission rate of the steganographic system (number of bits of secret message transmitted by covertext sent). Indeed, without use of error correcting code the failure probability of concealment of this system varies between 1/4 and 1/3. It depends on the covertexts distribution.

Shannon has shown that for a channel having a probability p of distortion of a symbol, the capacity of this channel is equal to $1 - H(p)$. On such a channel, one can reliably communicate with a corrector code with a rate close to $1 - H(p)$. Taking p = 1/4, the rate goes to 0.2 so covertexts 5 for a single secret message bit transmitted. their concealment algorithm runs in $O(\frac{1}{1-H(p)} \times n')$ plus time of error correcting code where n' is the length of secret message.

L. Reyzin and S. Russell [6] generalized the protocol proposed in [3] in order to be safe for steganographic distribution which has a small min-entropy. L. Reyzin and al. proceeded as follows: to hide a bit of secret message, it uses t covertexts instead of only one. By doing so, it increases the min-entropy of the covertexts distribution. his concealment algorithm runs in $O(|K| \times t \times n)$.

Algorithm 7 Rejection sampling [6]

Procedure RS

1: **Input**: *Key K1, Target y, iteration count, history h, number of covertexts t*
2: $i = 0$
3: **repeat**
4:     $h' = h$
5:     **for** $j = 1 \cdots t$ **do**
6:         $x_j \leftarrow \mathbf{S}(h)$
7:         $h' = h' || x_j$
8:     **end for**
9:     $x = x_1 x_2 \cdots x_t$
10:     **Increment** i
11: **until** $i = count$ or $F_{K1}(x) = y$
12: **Output:** x

Algorithm 8 Embedding procedure [6]

Procedure $S3.Embed$

1: **Input**: *Key K, hiddentext $m \in \{0,1\}^*$, history h, number of covertexts t*
2: **Parse** m as $m_1^1 || m_2^1 || \cdots || m_n^1$
3: **for** $i = 1 \cdots n$ **do**
4:     $c_i = RS(K, m_i, |K|, h, t)$
5:     $h = h || c_i$
6: **end for**
7: **Output:** $c_1 c_2 \cdots c_n$

Algorithm 9 Extracting procedure [6]

Procedure $\mathcal{S}3.Extract$

1: **Input**: $Key\ K$, $stegotexts\ c_1 c_2 \cdots c_n$
2: **for** $i = 1 \cdots n$ **do**
3:     $m_i = F_K(c_i)$
4: **end for**
5: $m = m_1 || m_2 || \cdots || m_n$
6: **Output:** m

N. Hopper et al. [5] proposed another steganographic system that no longer uses error correcting code. To hide a single bit, this system makes t copies of the secret message. And for each copy, it seeks a covertext in the channel whose image via a pseudo-random function is equal to the secret message by taking more than twice from the channel. the concealment and extracting algorithms run in $O(t)$ for one bit.

Algorithm 10 Embedding procedure [5]

Procedure $\mathcal{S}4.Embed$

1: **Input**: $Key\ K$, $hiddentext\ m \in \{0,1\}$, $history\ h$, $Synchronized\ N$
2: **for** $i = 1 \cdots t$ **do**
3:     $d_i, d_i' \leftarrow \mathbf{S}(h)$
4:     **if** $F_K(N+i, d_i) = m$ **then**
5:         $s_i = d_i$
6:     **else**
7:         $s_i = d_i'$
8:     **end if**
9:     $h = h || s_i$
10: **end for**
11: $N = N + t$
12: **Output:** $s_1 \cdots s_t$

Algorithm 11 Extracting procedure [5]

Procedure $\mathcal{S}4.Extract$

1: **Input:** $Key\ K$, $Synchronized\ N$, $stegotexts\ s_1 \cdots s_t$
2: $C = 0$
3: **for** $i = 1 \cdots t$ **do**
4:     $C = C + F_K(N, s_i)$
5:     **Increment** N
6: **end for**
7: **if** $C > t/2$ **then**
8:     $m = 1$
9: **else**
10:     $m = 0$
11: **end if**
12: **Output**: m

---

2   Our contribution

Hopper et al. [3] proposed a steganographic system whose transmission rate is 1 bit per document sent. In order to improve the safety of the system to expand the channel distributions that have small min-entropy, they proposed in [4] a system that reduces the transmission rate to 1 bit for 5 sent documents. Generalizing the system [3] to be applied to channels that have small min-entropy, Reyzin L. and S. Russell [6] propose a system which also reduces the transmission rate to 1 bit for t sent documents. Hopper et al. [5] proposed a system that has the same transmission rate as [6].

The systems proposed in [4, 5, 6] significantly reduce the transmission rate of one of [3]. These systems preserve probabilistic property of the concealment algorithm [3] and ask several sampling operations. The run-time of all these concealment algorithms is considerable: generally $O(\alpha.n)$.

We propose two steganographic systems for the communication by email with concealment algorithms are deterministic. To hide a secret message bit, our algorithms only perform one sampling operation and transmission rate is 1 bit document sent by reducing their execution time in $O(n)$. The safety of protocols is reduced to that of a pseudo-random function.

This article is organized as follows: In section 3, we give some definitions that support the solution developed in this work. In section 4, we present our steganographic protocols and we conclude in section 5.

3 Definitions

Formatting convention: Let $< a_1, a_2, \cdots, a_n >$ an array of elements. Two tables are said to be equal if the same index items are equals.

3.1 Channel

Let $\sum$ be a set of documents. A channel $\mathcal{C}$ is a function that takes as input a history $h \in \sum^*$ and provides a distribution probability $\mathcal{D}_h$.

The channel formalizes a normal communication between two entities. We define here a normal communication by email making use of the channel.

Let $\mathbb{A}$ be a set of array of addresses and $\mathcal{A}$ a uniform distribution on $\mathbb{A}$. We define an email by a triplet (d, adr, t) where $d \in \sum$, t the sent date of message and $adr \in \mathbb{A}$ an array of addresses containing the receivers' addresses.

Let $h = d_1 d_2 \cdots d_n$ be the history of already sent messages. A mail (d, adr, t) is said to be normal if d is drawn at random from the channel and adr is chosen at random in $\mathcal{A}$.

3.2 Steganographic system

Let $\mathbb{A}$ be a set of array of addresses and $\sum$ a set of documents. A steganographic protocol or steganographic system is a pair of algorithms $\mathcal{S} = (\mathcal{S}.Embed, \mathcal{S}.Extract)$:

- $\mathcal{S}.Embed$ *takes as input a key* $K \in \{0,1\}^k$, *a string* $m \in \{0,1\}^*$ *(hiddentext), a history* h *and has access to* **S**. $\mathcal{S}.Embed(k, m, h)$ *returns a sequence of mails* $s_1 s_2 \cdots s_n$ *where* $s_i = (d_i, adr_i, t_i)$.

- $\mathcal{S}.Extract$ *takes as input a key* K, *two sequences of messages* $s_1 s_2 \cdots s_{n_1}$ *and* $s_1' s_2' \cdots s_{n_2}'$. $\mathcal{S}.Extract$ *returns the secret message* m

Reliability

$\mathcal{S}.Embed$ and $\mathcal{S}.Extract$ must satisfy the following relationship:

$$\forall m : \Pr[\mathcal{S}.Extract(K, \mathcal{S}.Embed(K, m, h)) = m] = 1$$

3.3 Security

Intuitively we require, according to the security of steganographic system, that no effective adversary G can distinguish mails returned by the concealment algorithm from normal mails. We assume that G knows distributions $\mathcal{D}_h$ and $\mathcal{A}$. We allow the adversary G to have access to $\mathcal{S}.Embed$ and h (communication history between Alice and Bob), and select a message m. Only the key is not known by G. We model an attack against a steganographic system as a game played by a passive adversary as: G has access to the oracle **M** that can be either:

- ST. The oracle ST has a uniformly chosen key K and responds to requests (m, h) with a mail drawn sequence $\mathcal{S}.Embed(K, m, h)$.

- CT. The CT oracle also has a uniformly chosen key K and respond to requests (m, h) with a sequence of normal mails $s_1 s_2 \cdots s_n$ where $s_i = (d_i, adr_i, t_i)$, $d_i \in \mathcal{D}_{h d_1 d_2 \ldots d_{i-1}}$ for $1 \leq i \leq n$. adr_i is randomly selected in $\mathcal{A}$. n is the number of mails returned by $\mathcal{S}.Embed(K, m, h)$.

After interaction with its oracle, G sets out a bit which represents his assumption about the type of **M**. He puts 1 to say that **M** is of type ST and 0 otherwise. We define the advantage of G against a steganographic system $\mathcal{S}$ for a channel $\mathcal{C}$ by:

$$\mathbf{Adv}_{\mathcal{S},\mathcal{C}}^{SS}(G) = |\Pr_{K \leftarrow \mathcal{K}}[G^{\mathbf{M}=ST} = 1] - \Pr_{K \leftarrow \mathcal{K}}[G^{\mathbf{M}=CT} = 1]|$$

where the probability is taken as the random effect of ST, CT and the choice of G. We define the insecurity of $\mathcal{S}$ by:

$$\mathbf{InSec}_{\mathcal{S},\mathcal{C}}^{SS}(t,q,l) = \max_{G \in \mathcal{G}(t,q,l)} \{\mathbf{Adv}_{\mathcal{S},\mathcal{C}}^{SS}(G)\}$$

where $\mathcal{G}(t,q,l)$ is the set of all the adversaries that makes at most q queries to the oracle, totaling at most l bits (hiddentexts) and runs at most in t steps.

3.4 Pseudorandom function

Let $\mathcal{F} = \{F_K\}_{K \in \{0,1\}^n}$ be a family of functions all with the same domain and co-domain. Let A be a probabilistic adversary with access to **Fn** sampling oracle. The prf-advantage of A to $\mathcal{F}$ is:

$$\mathbf{Adv}_F^{prf}(A) = |\Pr_{K \leftarrow \{0,1\}^n}[A^{\mathbf{Fn}=F_K} = 1] - \Pr[A^{\mathbf{Fn}=f} = 1]|$$

where f is a random function of the same domain and co-domain F_K. The insecurity of $\mathcal{F}$ is given by the following formula:

$$\mathbf{InSec}_F^{prf}(t,q) = \max_{A \in \mathcal{A}(t,q)} \{\mathbf{Adv}_F^{prf}(A)\}$$

where $\mathcal{A}(t,q)$ denotes the set of opponents performing at most t steps and makes at most q queries to the oracle.

4 Our protocols

We present in this section our steganographic protocols that conceal several bits. The security of these protocols is based on the difficulty to solve a cryptographic problem: the break of a pseudo-random function.

Each of these protocols uses two primitives: *extractDocument* which takes as input a mail and returns the document in this email and *extractAddresses* which takes as input a mail and returns an array of the destination email addresses.

We assume also that Bob owns two addresses: *address1* and *address2* known by Alice and Eve.

4.1 Steganographic protocol 1

4.1.1 Secret steganographic state for one bit

Alice and Bob share a channel $\mathcal{C}$. $\mathcal{F}$ is a pseudo-random function where $F_K : \{0,1\}^d \times \sum \rightarrow \{0,1\}$. Alice and Bob possess a secret key $K \in \{0,1\}^k$ and are synchronized by a counter $N \in \{0,1\}^n$. Let $\mathbb{A} = \{< address1 >, < address2 >\}$ be a set of array of addresses and $\mathcal{A}$ an uniform distribution of $\mathbb{A}$. The following algorithms allow Alice and Bob to hide and extract a bit of secret message.

Algorithm 12 Embedding procedure for one bit

Procedure $\mathcal{S}.EmbedOneBit$

1: **Input**: *Key K, Hiddentext* $m \in \{0,1\}$, *Synchronized N, history h, Sent date t*
2: $d \leftarrow \mathbf{S}(h)$
3: **if** $(F_K(N,d) = m)$ **then**
4:    $s = (d, < address1 >, t)$
5: **else**
6:    $s = (d, < address2 >, t)$
7: **end if**
8: **Increment** N
9: $h = h||d$
10: **Output:** s

Algorithm 13 Extracting procedure for one bit

Procedure $\mathcal{S}.ExtractOneBit$

1: **Input:** *Key K, Synchronized N, mail s*
2: $d = \text{extractDocument}(s)$
3: $m = F_K(N, d)$
4: **if** $(extractAddresses(s) = <address2>)$ **then**
5:     $m = \overline{m}$
6: **end if**
7: **Increment** N
8: **Output:** m

    The idea behind this algorithm is simple. Suppose Alice and Bob can communicate using two envelopes of different colors (white and brown). Alice, to send a message to Bob, draws a document d from the channel, evaluates it by the function $F_K(N, \cdot)$. If the result $F_K(N, d)$ is equal to the bit of message she wants to hide, Alice sends the document d in the white envelope otherwise it's sent in the brown envelope. When Bob receives the document d in a brown envelope, he takes the complement of the secret message provided by the document otherwise he takes the bit as provided.

Run-time: The complexity in time is constant: $T = O(1)$.

Reliability: Our protocol uses deterministic algorithms, therefore we can easily find all the bits embedded in the mails.

Insecurity: We show here that the task of an adversary to distinguish, between the distribution of mails returned by the concealment algorithm from normal mails, is more difficult than an adversary of a pseudo-random function. The proof of the theorem below is similar to the ones of [4, 5].

First, we will show that the distribution of documents returned by the concealment algorithm is equal to that of the channel. Then we also show that when replacing the function $F_K(\cdot, \cdot)$ by a random function, the distribution of addresses returned by concealment algorithm is equal to $\mathcal{A}$. And later, we will build an opponent against a pseudorandom function that has the same advantage an adversary of a steganographic system.

Lemma 1: *The probability of a document received from channel, returned by $\mathcal{S}.EmbedOneBit(K, m, N, h)$ is equal to the probability of this document in the channel $\mathcal{D}_h$.*

Proof: Let d_i document content in the email back $\mathcal{S}.EmbedOneBit(K, m, h)$. The probability that d_i is put in output only depends on its drawn in the channel conditioned by history h. Thus

$$\Pr_{(d_i, adr_i, t_i) \leftarrow \mathcal{S}.EmbedOneBit(K, m, N, h)}[d_i] = \Pr_{\mathcal{D}_h}[d_i]$$

Lemma 2: When the function $F_K(\cdot, \cdot)$ is replaced by a random function f, for all $h \in \sum^*$ the probability of address in the mail returned by $\mathcal{S}.EmbedOneBit(K, m, N, h)$ is equal to her probability in $\mathcal{A}$.

Proof: Let m be the bit of secret message. The probability that sends a document to the address $<adresseI> \in \mathbb{A}$ depends on the evaluation of the document and the counter N in the function f. And as the counter is incremented every sent document the entrance (N, d) will always be different for the opponent even if a document is sent more than once.

$$\Pr[<adresseI>] = \Pr[f(N, d) = m]$$

$$= \frac{1}{2}$$

Theorem 1:

$$\mathbf{InSec}_{\mathcal{S}, \mathcal{C}}^{SS}(t, q, q) \leq \mathbf{InSec}_F^{prf}(t + qO(1), q)$$

Proof: Let $G \in \mathcal{G}(t, q, l)$ an adversary for $\mathcal{S}$. We construct an opponent A for a pseudorandom function $\mathcal{F}$ with same advantage G. The construction algorithm is given below:

- $A^{\mathbf{Fn}}$ works by executing G and intercepting all queries sends his oracle $\mathbf{M}$,

- To respond these demands, $A^{\mathbf{Fn}}$ simulates concealment algorithm: $\mathcal{S}.EmbedOneBit$ using the oracle $\mathbf{Fn}$ instead of F_K and gets a result c; and starts with G with c as response to its request.

- When G stops with output b, A also puts b output.

Clearly, where $\mathbf{Fn} = F_K$, $A^{\mathbf{Fn}}$ perfectly simulates the oracle ST of G, as follows:

$$\Pr_{K \leftarrow \{0,1\}^k}[A^{\mathbf{Fn}=F_K} = 1] = \Pr_{K \leftarrow \mathcal{K}}[G^{\mathbf{M}=ST} = 1] \tag{1}$$

By the previous Lemma 1 and Lemma 2, when $\mathbf{Fn}$ is a random function f, $A^{\mathbf{Fn}}$ perfectly simulates the CT oracle G:

$$\Pr[A^{\mathbf{Fn}=f} = 1] = \Pr_{K \leftarrow \mathcal{K}}[G^{\mathbf{M}=CT} = 1] \tag{2}$$

Subtracting (2) from (1) we obtain:

$$\mathbf{Adv}_F^{prf}(A) = \mathbf{Adv}_{\mathcal{S},\mathcal{C}}^{SS}(G)$$

The theorem follows by definition of insecurity.

4.1.2 Secret steganography state for several bits

We now present our first protocol that allows steganography to conceal several bits. The algorithms below allow Alice and Bob to embed and extract more bits of secret messages.

Algorithm 14 Embedding procedure for multiple bits

Procedure $\mathcal{S}.Embed$

1: **Input**: *Key K, Hiddentext* $m \in \{0,1\}^*$,*Synchronized N, history h, Sent date t*
2: **Parse m as** $m_1^1 || m_2^1 || \cdots || m_n^1$
3: **for** $i = 1 \cdots n$ **do**
4:    $s_i = \mathcal{S}.EmbedOneBit(K, m_i, N, h, t)$
5:    $d_i = extractDocument(s_i)$
6:    $h = h || d_i$
7:    **Increment** N
8:    $t = t + 1$
9: **end for**
10: **Output:** $s_1 s_2 \cdots s_n$

Algorithm 15 Extracting procedure for multiple bits

Procedure $\mathcal{S}.Extract$

1: **Input**: *Key K, Synchronized N, Mails address1* $s_1 s_2 \cdots s_{n_1}$, *Mails address2* $s_1' s_2' \cdots s_{n_2}'$
2: $s =$ merge the two mails sequences sorted in ascending order of sent date.
3: **for** $i = 1 \cdots n$ **do**
4:    $m_i = \mathcal{S}.ExtractOneBit(K, N, s_i)$
5:    **Increment** N
6: **end for**
7: **Output**: $m_1 || m_2 || \cdots || m_n$

The sequences $s_1 s_2 \cdots s_{n_1}$ and $s_1' s_2' \cdots s_{n_2}'$ are the mails contained respectively in the mail boxes *address*1 and *address*2.

Run-time: The concealment algorithm runs in $O(n)$. Concerning the extraction algorithm, note that mails received in a mail box are recovered in the sending order. This means that the two sequences are sorted by date of mailing. The statement 2 of the extraction algorithm is there to merge these two sequences in a sorted sequence. This operation is performed in at most $O(n)$. Thus, the extraction algorithm runs in $O(n) + O(n)$ which is equal to $O(n)$.

4.2 Steganography Protocol 2

4.2.1 Secret steganography to state for a bit

The idea of this protocol is to do a broadcast with destination addresses. The order of the addresses in the broadcast is very important. This order determines whether the recipient must take the extracted secret message bit of the document or its complement. From the two recipient addresses, one can have only two types of diffusion address with respect of to the order. So $\mathbb{A} = \{< address1, address2 >, < address2, address1 >\}$. Let $\mathcal{A}$ be a uniform distribution on $\mathbb{A}$. We only need to consider mails sent one address recipient's to extract the secret message.

Algorithm 16 Embedding procedure for one bit

Procedure $\mathcal{S}'.EmbedOneBit$

1: **Input:** *Key K, Hiddentext $m \in \{0, 1\}$, Synchronized N, history h, Sent date t*
2: $d \leftarrow \mathbf{S}(h)$
3: **if** $(F_K(N, d) = m)$ **then**
4:    $s = (d, < address1, address2 >, t)$
5: **else**
6:    $s = (d, < address2, address1 >, t)$
7: **end if**
8: **Increment** N
9: $h = h || d$
10: **Output:** s

Algorithm 17 Extracting procedure for one bit

Procedure $\mathcal{S}'.ExtractOneBit$

1: **Input:** *Key K, Synchronized N, mail s*
2: $d = \text{extractDocument}(s)$
3: $m = F_K(N, d)$
4: **if** $(extractAddresses(s) = < address2, address1 >)$ **then**
5:    $m = \overline{m}$
6: **end if**
7: **Increment** N
8: **Output:** m

Run-time: The execution time of this protocol is practically constant for concealment and extraction algorithms and depends on the size of the secret message that is obviously 1: $T = O(1)$.

Insecurity: The security of this protocol is exactly the same as the one expressed in the theorem 1.

4.2.2 Secret steganography state for several bits

Algorithm 18 Embedding procedure for multiple bits

Procedure $\mathcal{S}'.EmbedMultiBits$

1: **Input:** *Key K, Hiddentext $m \in \{0, 1\}^*$, Synchronized N, history h, Sent date t*
2: **Parse m as** $m_1^1 || m_2^1 || \cdots || m_n^1$
3: **for** $i = 1 \cdots n$ **do**
4:    $s_i = \mathcal{S}'.EmbedOneBit(K, m_i, N, h, t)$
5:    $d_i = extractDocument(s_i)$
6:    $h = h || d_i$
7:    **Increment** N
8:    $t = t + 1$
9: **end for**
10: **Output:** $s_1 s_2 \cdots s_n$

Algorithm 19 Extracting procedure for multiple bits

Procedure $\mathcal{S}'.ExtractMultisBits$

1: **Input**: *Key K*, *Synchronized N*, *Mails* $s_1 s_2 \cdots s_n$
2: **for** $i = 1 \cdots n$ **do**
3:    $m_i = \mathcal{S}'.ExtractOneBit(K, N, s_i)$
4:    **Increment** N
5: **end for**
6: **Output**: $m_1 || m_2 || \cdots || m_n$

Run-time: $T = O(n)$.

5   Conclusion

In this work, we were interested in the reducing of the time complexity and the number of documents sent to hide a secret message. Our protocols use a constant time to hide and extract a bit of secret message and run in $O(n)$ to hide and extract n bits of a secret message.

Acknowledgements

References

[1] G. J. Simmons. The prisoners' problem and the subliminal channel. In David Chaum, editor, Advances in Cryptology: Proceedings of Crypto 83, pages 51–67. Plenum Press, New York and London, 1984, 22–24 August 1983.

[2] C. Cachin. An information-theoretic model for steganography. In Second Internation Workshop on Information Hiding, volume 1525 of Lecture Notes in Computer Science, pages 306–316, 1998.

[3] N. Hopper, J. Langford, and L. von Ahn. Provably secure steganography. In Moti Yung, editor, Advances in Cryptology—CRYPTO 2002, Lecture Notes in Computer Science. Springer-Verlag, 18–22 August 2002.

[4] N. Hopper, J. Langford, and L. von Ahn. Provably secure steganography. Technical Report CMU-CS-02-149, School of Computer Science, Carnegie Mellon University, 2002.

[5] N.J. Hopper, L. von Ahn, J. Langford, Provably secure steganography. IEEE Trans. Comput. 58(5),662–676 (2009).

[6] L. Reyzin and S. Russell, "Simple Stateless Steganography," Cryptology ePrint Archive, Report 2003/093, http://eprint.iacr.org/, 2003.

[7] N. Dedìc, G. Itkis, L. Reyzin, and S. Russell, "Upper and Lower Bound on Black-Box Steganography", proc. Second Theory of Cryptography conf. (TTC' 05), J. Kilian, ed., pp. 227-244, 2005.

[8] Ross J. Anderson and Fabien A. P. Petitcolas. On The Limits of Steganography. IEEE Journal of Selected Areas in Communications, 16(4). May 1998.

[9] D. Kahn, The Codebreakers. New York: Macmillan, 1967.

[10] R.J. Anderson, "Stretching the Limits of Steganography," Proc. First Int'l Workshop Information Hiding (IH '96), pp. 39-48, 1996.